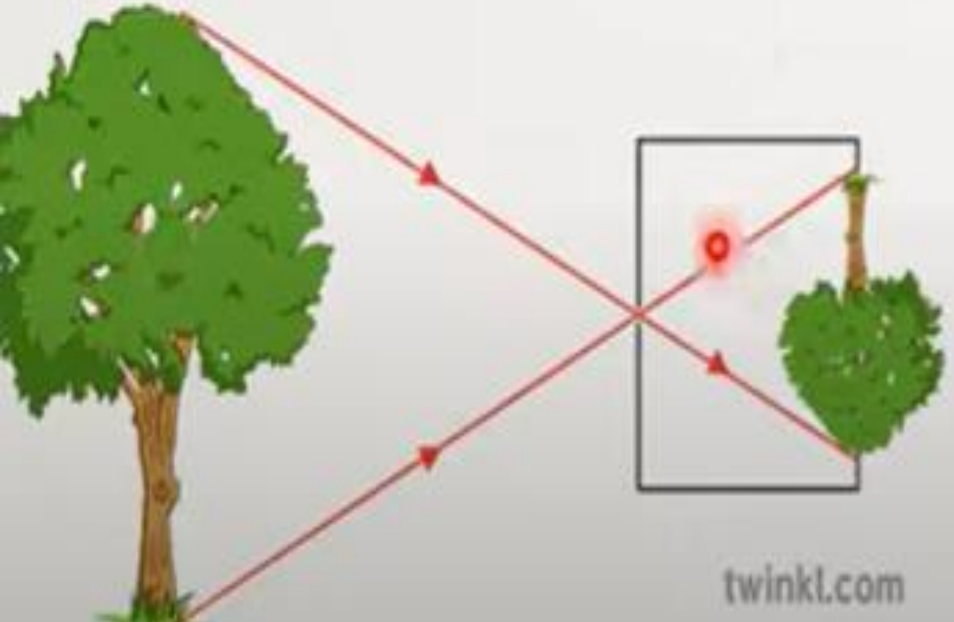
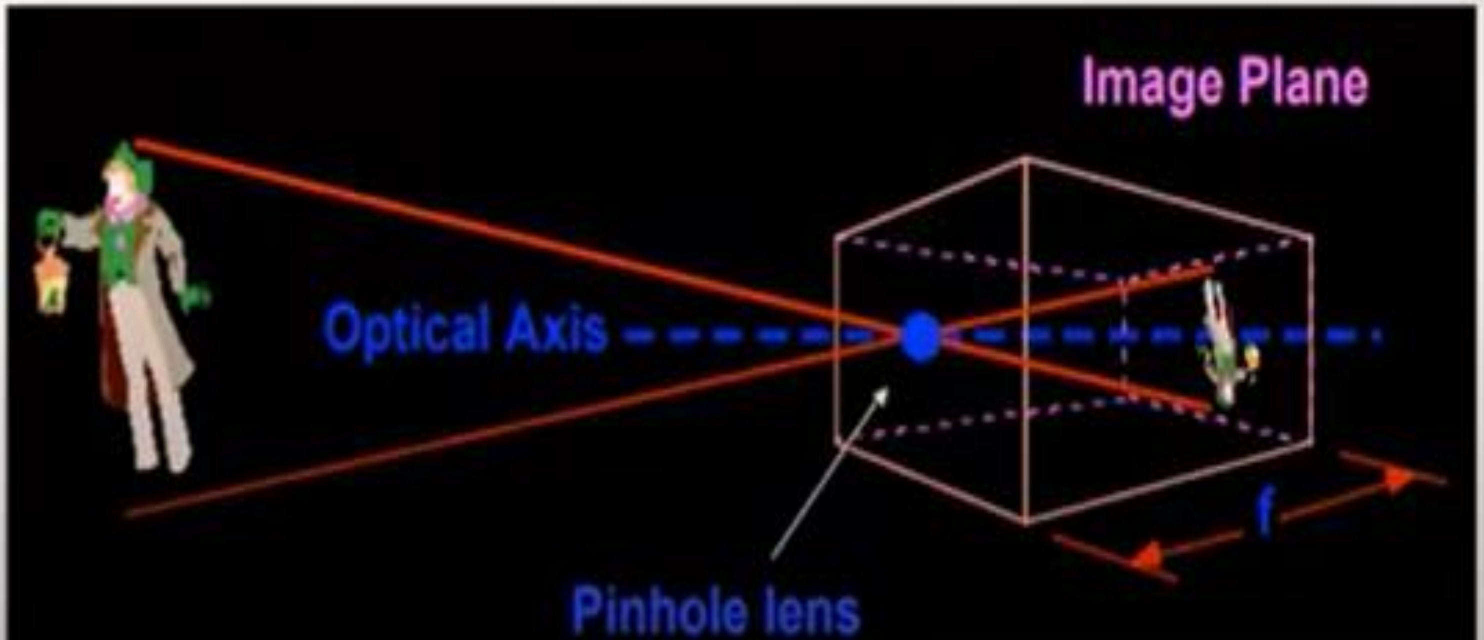
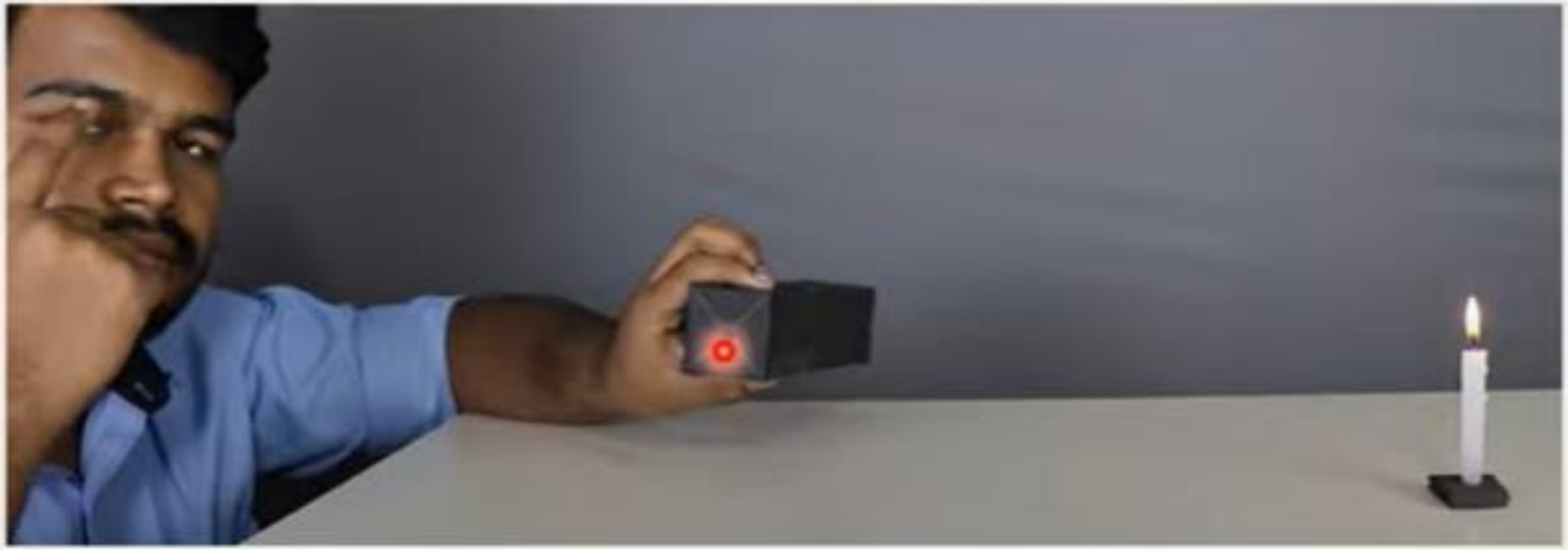
Pinhole Camera Model

- A pinhole camera is "a simple camera without a lens and with a single small aperture." Many pinhole cameras are as a simple as a box with a hole in the side.





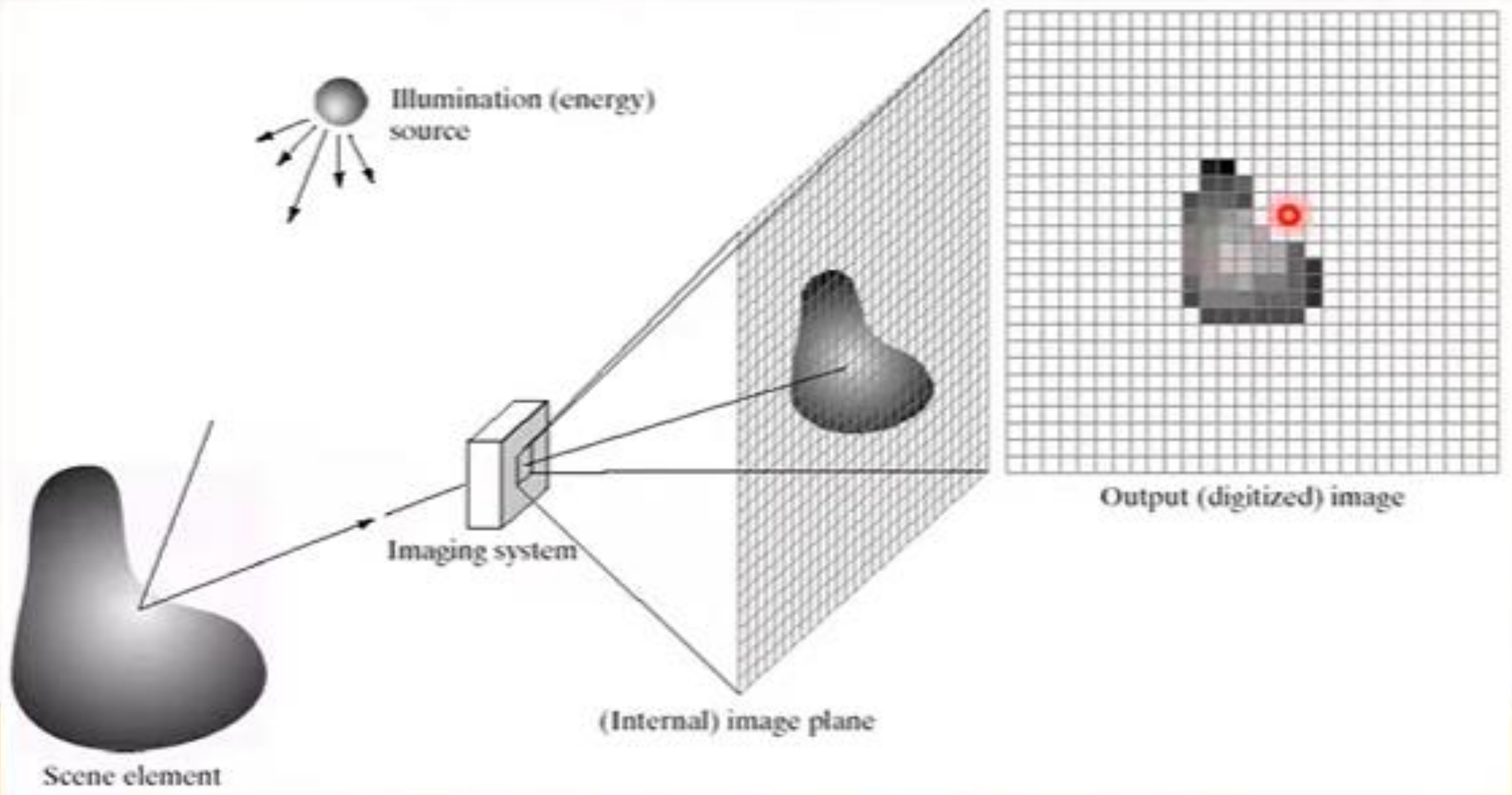


- With a pinhole camera, this image is usually upside down and varies in clarity. Some people use a pinhole camera to study the movement of the sun over time (Solargraphy). A type of pinhole camera is often used to view an eclipse. Another type of pinhole camera, the camera obscura, was once used by artists.
- The device allowed the artist to view a scene through a different perspective. The artist would point the lens of the camera at the still life scene they wanted to paint. The camera would frame the image in smaller perspective thus allowing the artist to see the scene as it might appear painted. (Of course, this would eventually lead to photography). The person using a camera obscura might even trace the image on a piece of paper and achieve a very accurate copy of the scene.

Image Digitization

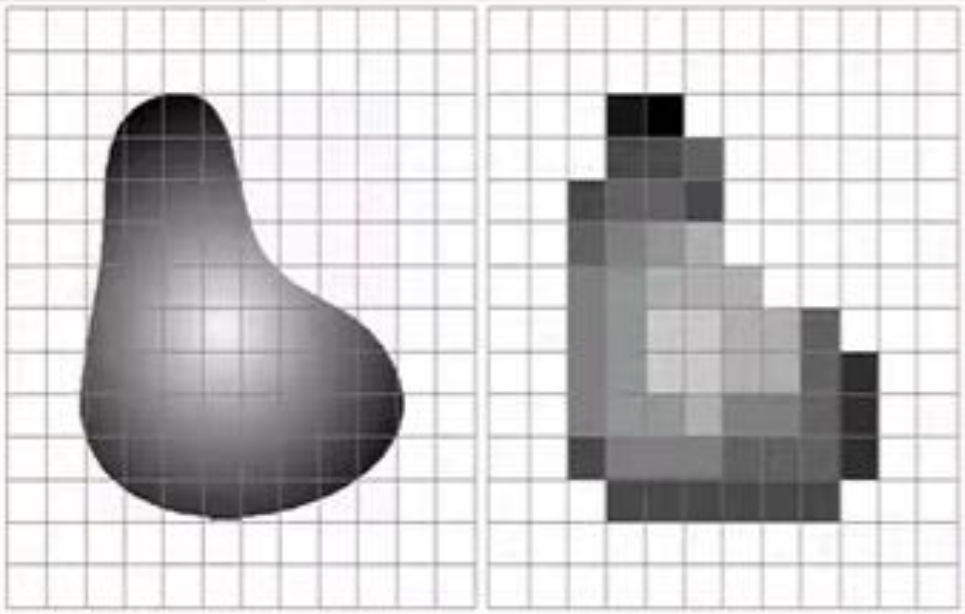
What is a Digital Image?

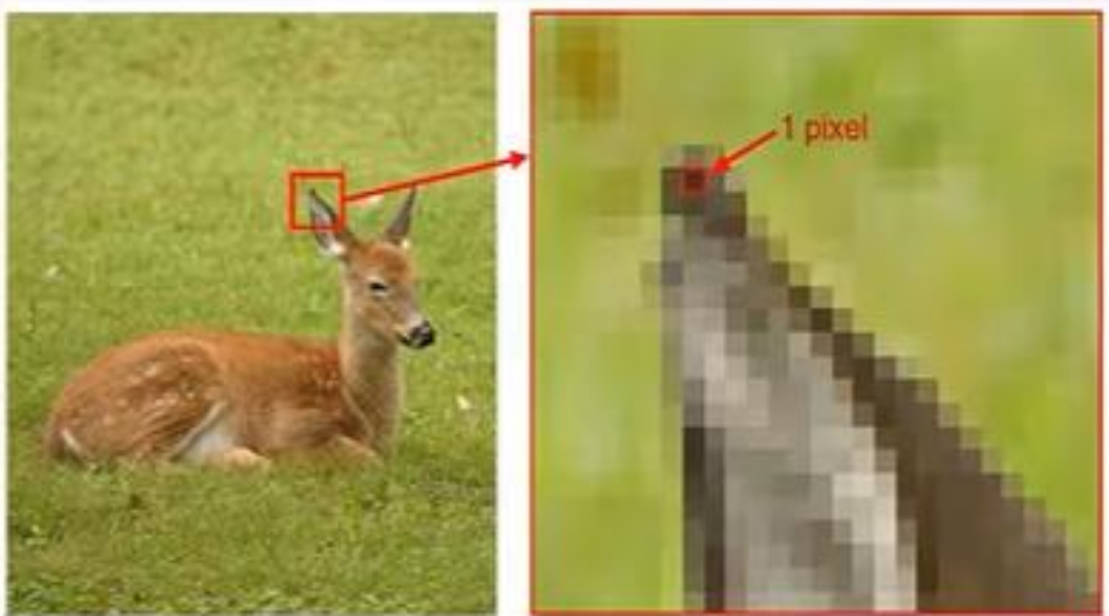
A **digital image** is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels



What is a Digital Image? (Cont.)

- Pixel values typically represent gray levels, colors, heights, opacities etc.
- **Remember** *digitization* implies that a digital image is an *approximation* of a real scene





What is a Digital Image? (Cont.)

Common image formats include:

- 1 sample per point (B&W or Grayscale)
- 3 samples per point (Red, Green, and Blue)
- 4 samples per point (Red, Green, Blue, and “Alpha”, a.k.a. Opacity)

Image Representation

Image Representation (Cont.)

- **Image as a matrix**
  - The simplest way to represent the image is in the form of a matrix.
  - In fig. 6, we can see that a part of the image, i.e., the clock, has been represented as a matrix. A similar matrix will represent the rest of the image too.

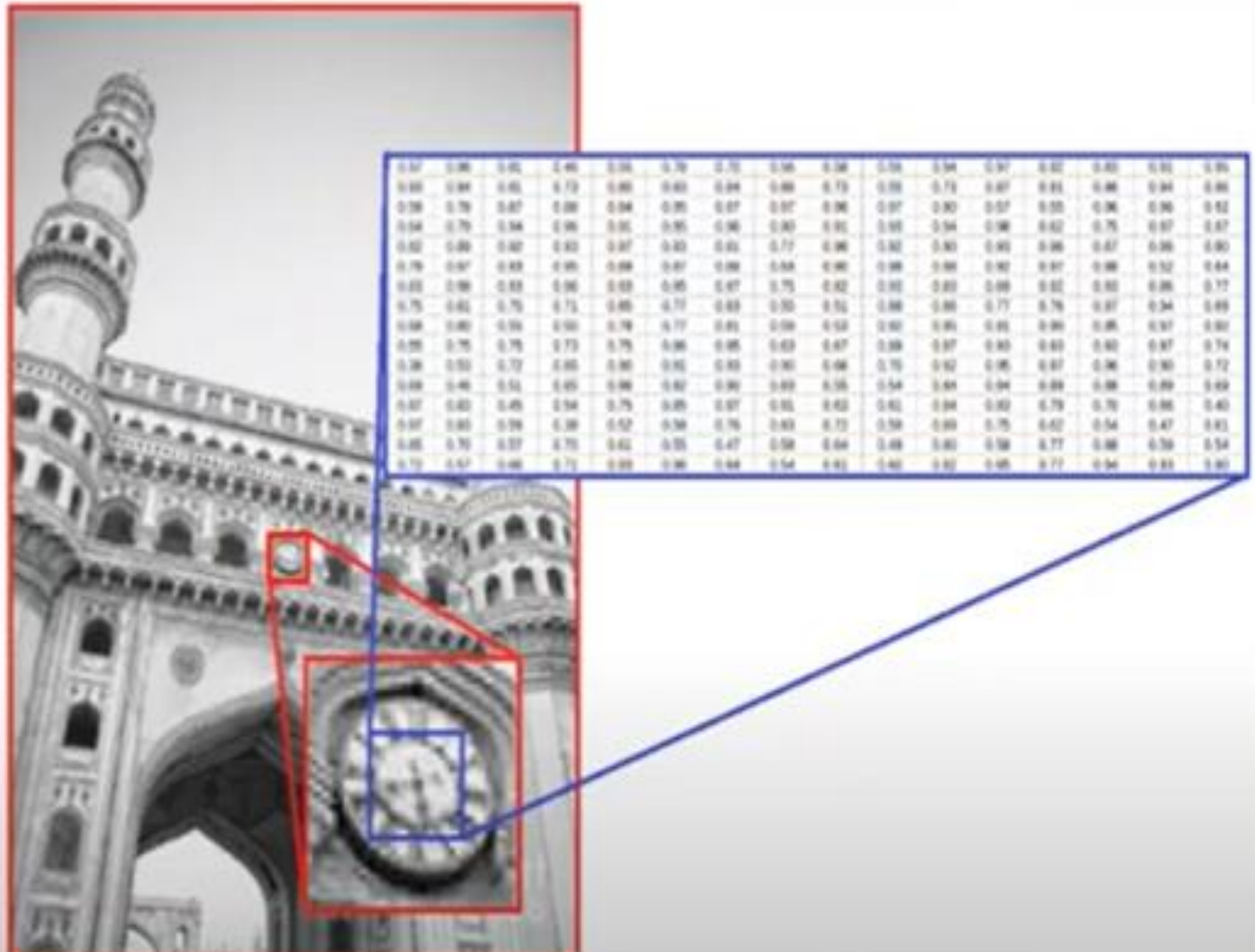


Image Representation (Cont.)

- It is commonly seen that people use up to a byte to represent every pixel of the image. This means that values between 0 to 255 represent the intensity for each pixel in the image where 0 is black and 255 is white. For every color channel in the image, one such matrix is generated.

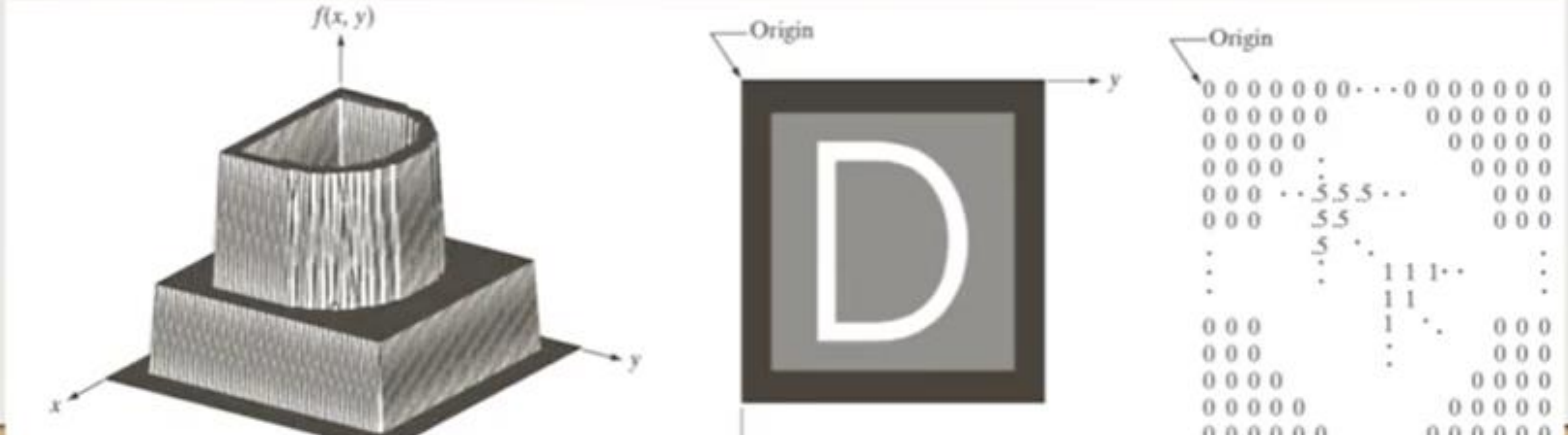


Image Representation (Cont.)

- Image as a function
  - An image can also be represented as a function. An image (grayscale) can be thought of as a function that takes in a pixel coordinate and gives the intensity at that pixel.
  - It can be written as function $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ that outputs the intensity at any input point (x,y). The value of intensity can be between 0 to 255 or 0 to 1 if values are normalized

Image Geometric/Spatial Transformation

Image Geometric/Spatial Transformation

- Image geometric that means changing the geometry of an image.
- Geometric transforms permit the elimination of geometric distortion that occurs when an image is captured.
- A spatial transformation of an image is a geometric transformation of the image coordinate system.
- In spatial transformation each point (x,y) of image A is mapped to a point (u, v) in a new coordinate system.

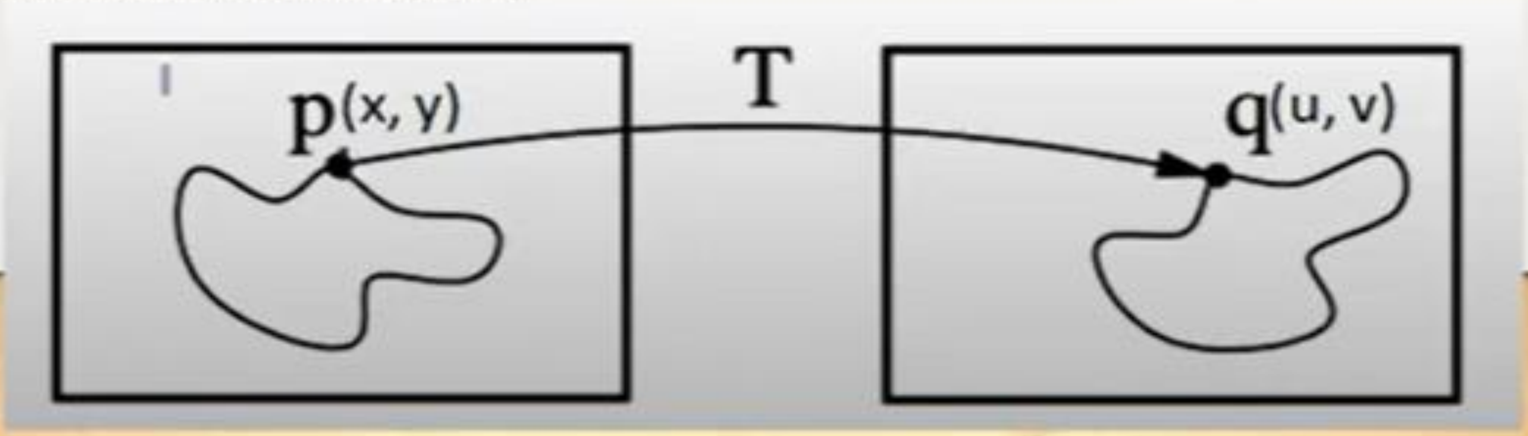
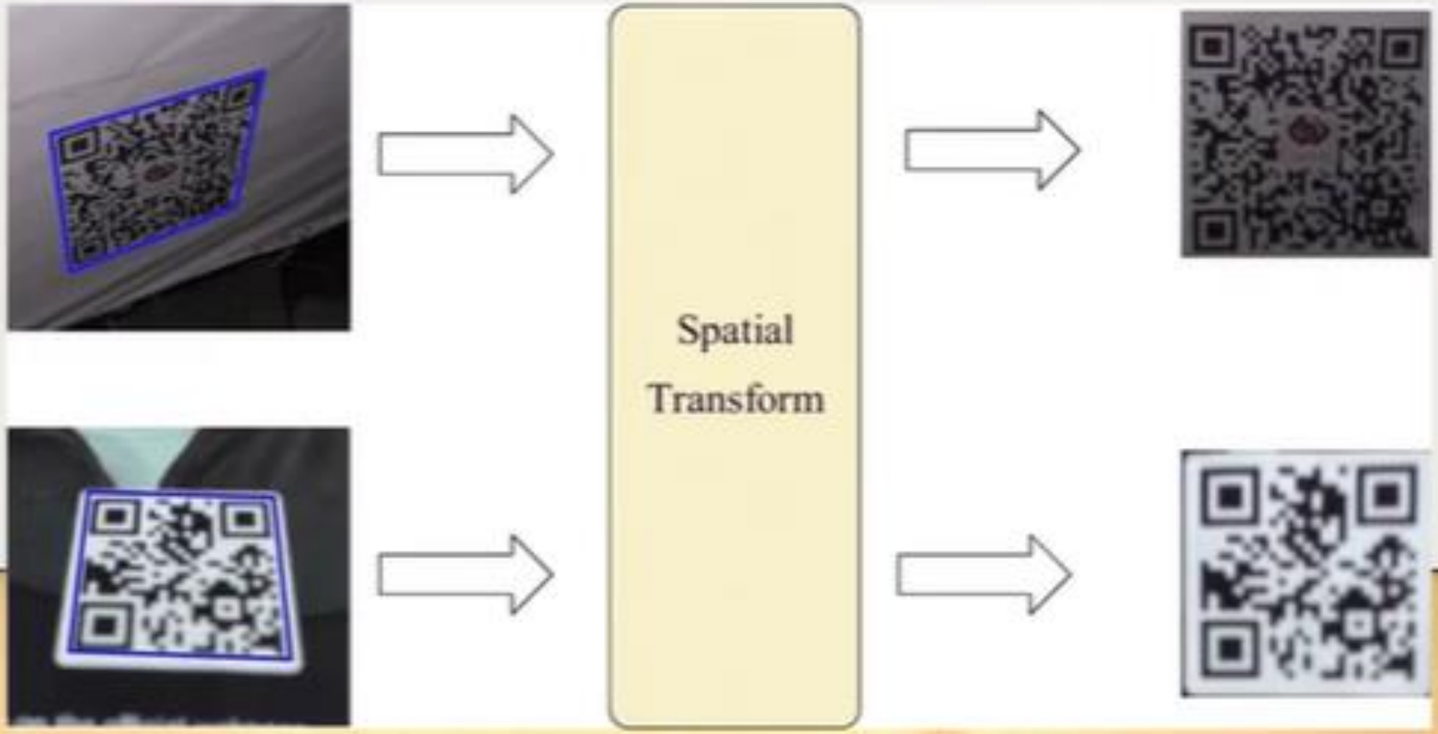


Image Geometric/Spatial Transformation

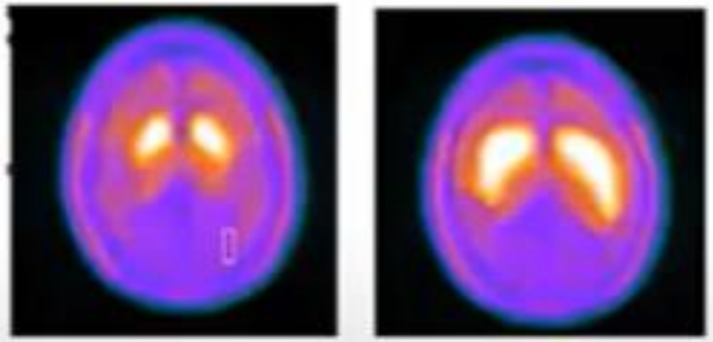
- Why it is used?
  - Some person clicking the pictures of the same place at different times of the day and year to visualize the changes. every time he clicks the picture, it's not necessary that he clicks the picture at the exact same angle. So for better visualization, he can align all the images at the same angle using geometric transformation.



Why geometric transformation in required?

- Image registration is the process of transforming different sets of data into one coordinate system.

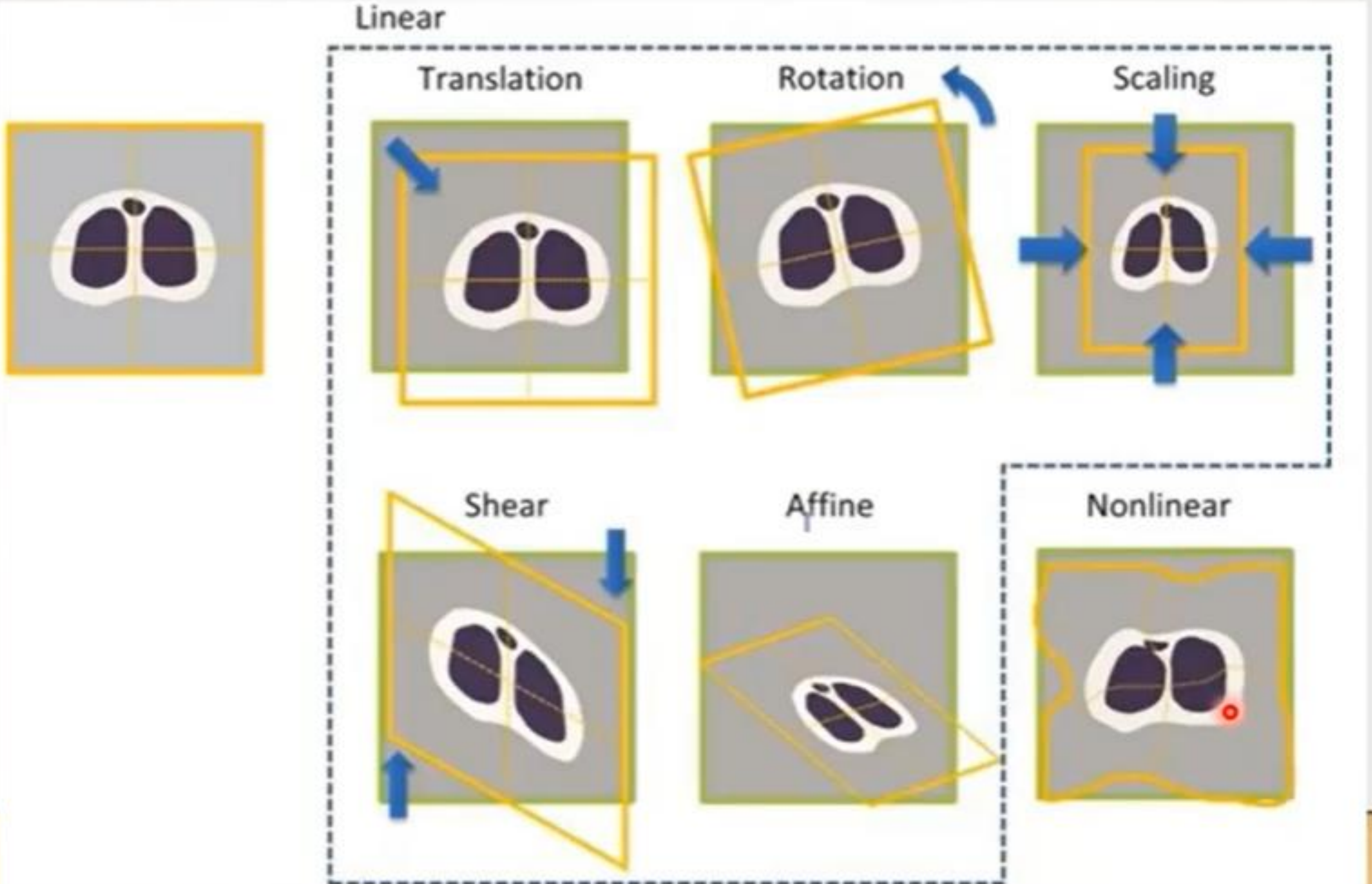
from different sensors


at different time


from different depths


from different viewpoints


from different viewpoints

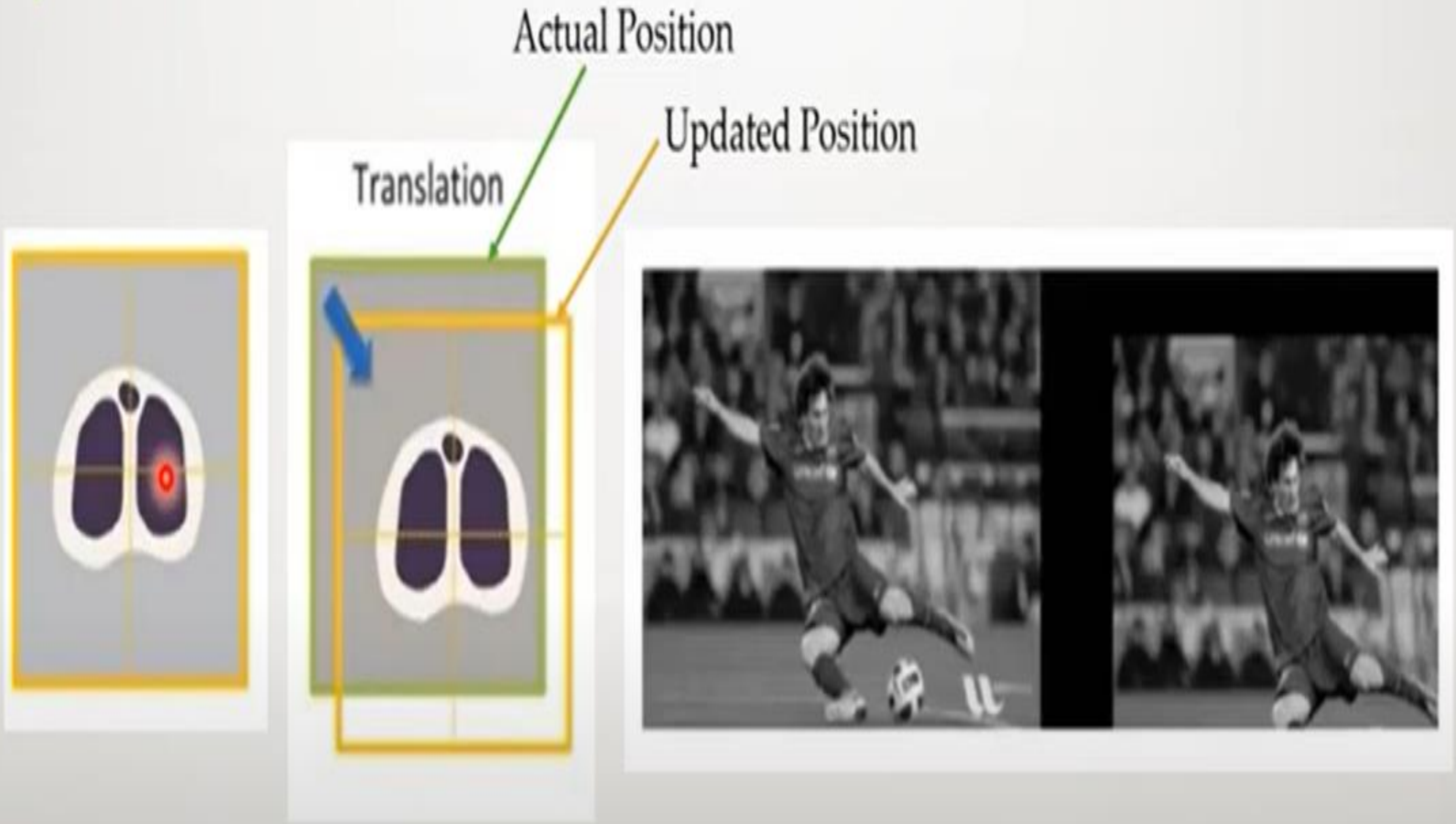

Linear
Translation
Rotation
Scaling
Shear
Affine
Nonlinear

Types of Geometric Transformation
1. Translation

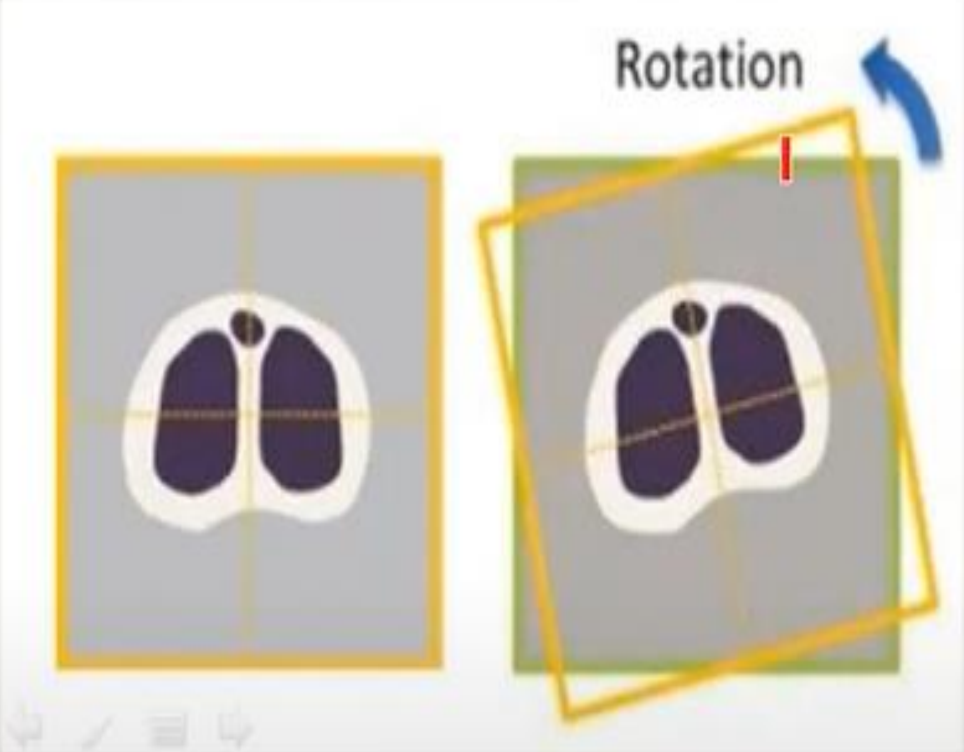
- Translation is the shifting of the object's location. If you know the shift in (x, y) direction, let it be, you can create the transformation matrix as follows:

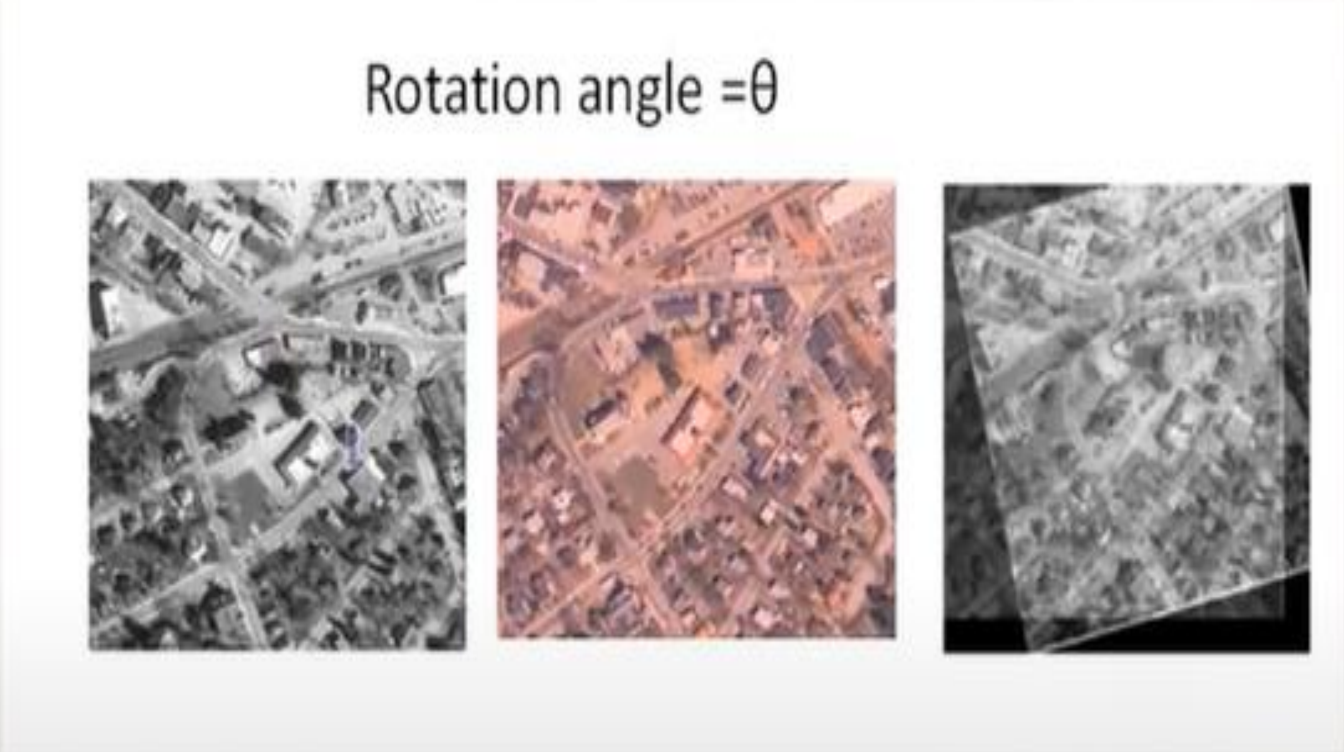
$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$


Actual Position
Updated Position
Translation

Types of Geometric Transformation
2. Rotation (Cont.)





Types of Geometric Transformation
2. Rotation (Cont.)

- This technique rotates an image by a specified angle and by the given axis or point.
- The points that lie outside the boundary of an output image are ignored. Rotation about the origin by an angle θ is given by,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$u = x\cos\theta + y\sin\theta$$
$$v = -x\sin\theta + y\cos\theta$$

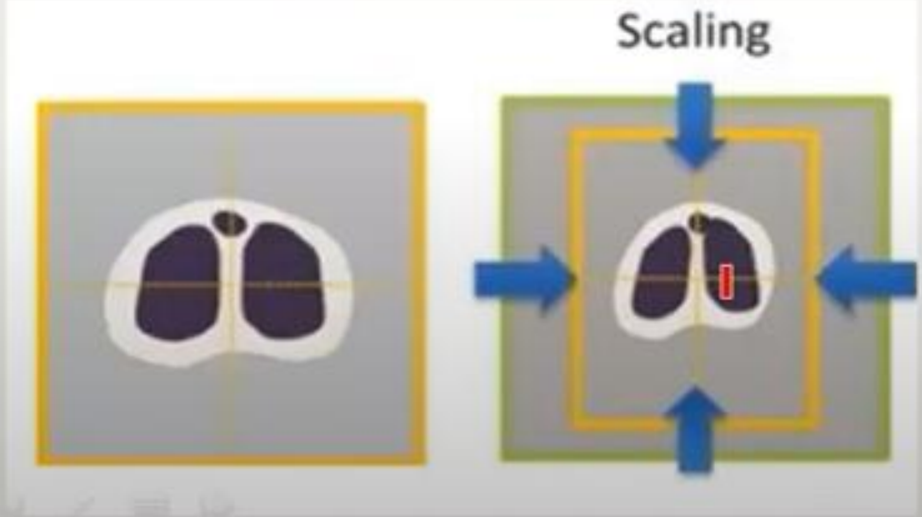
Types of Geometric Transformation
3. Scaling

- Scaling means resizing an image which means an image is made bigger or smaller in x/y direction.
- We can resize an image in terms of scaling factor.

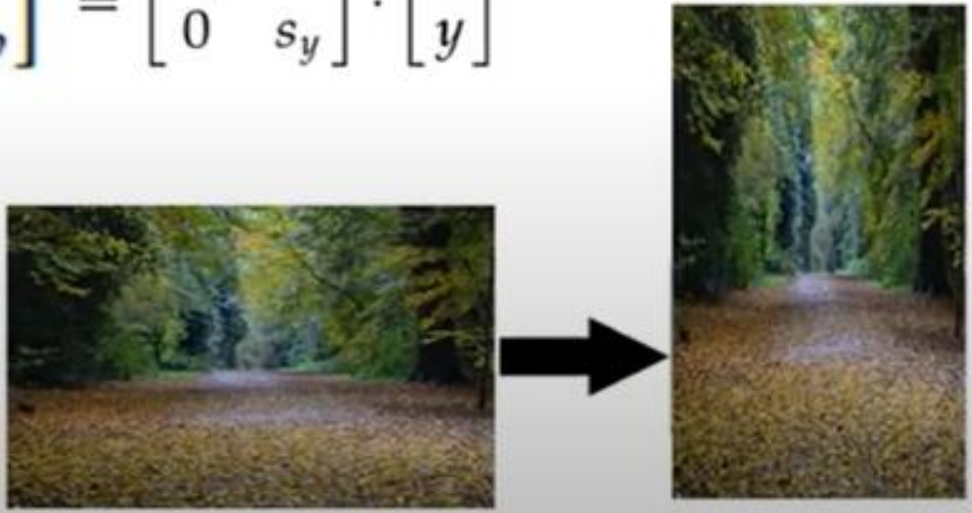
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Types of Geometric Transformation
3. Scaling (Cont.)

If we have an image of size (300 x 400) and we want to transform it into an image of shap
(600 x 200).
The scaling in x- direction will be : 600/300 = 2. (we denote it as Sx = 2)
Similarly Sy = 200/400 = 1/2.



$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$



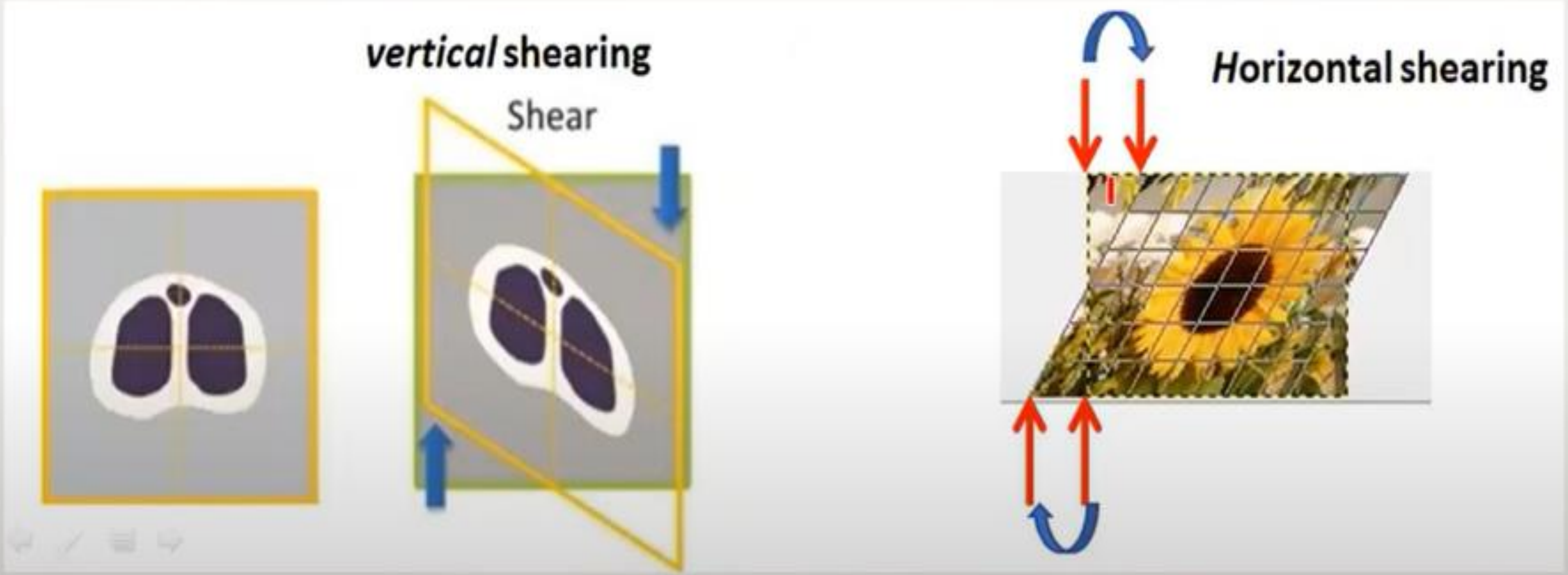
(x,y) = (300, 400)

- 300 row and 400 col.
- 300= height 400= width

Types of Geometric Transformation
4. Shearing

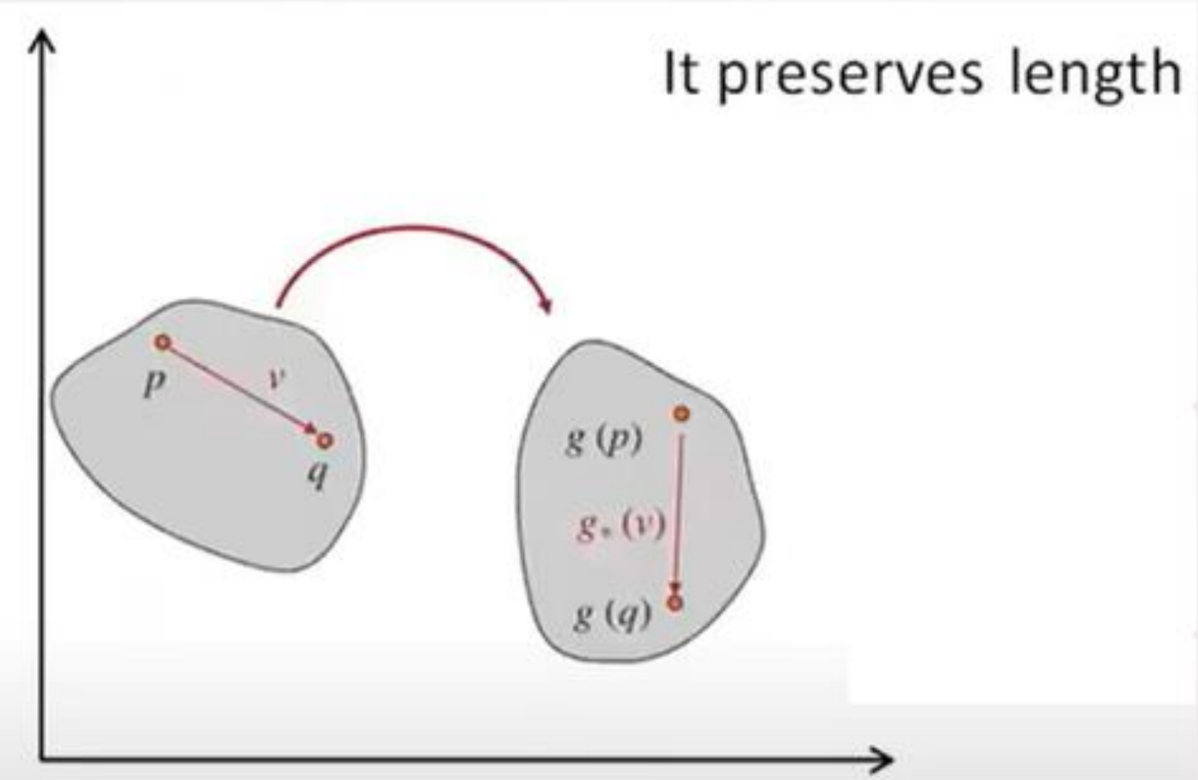
- Shearing an image means shifting the pixel values either horizontally or vertically.
- Basically, this shits some part of an image to one direction an other part to some other direction. Horizontal shearing will shift the upper part to the right and lower part to the left.
- Here you can see in gif. That upper part has shifted to the right and the lower part to the left.

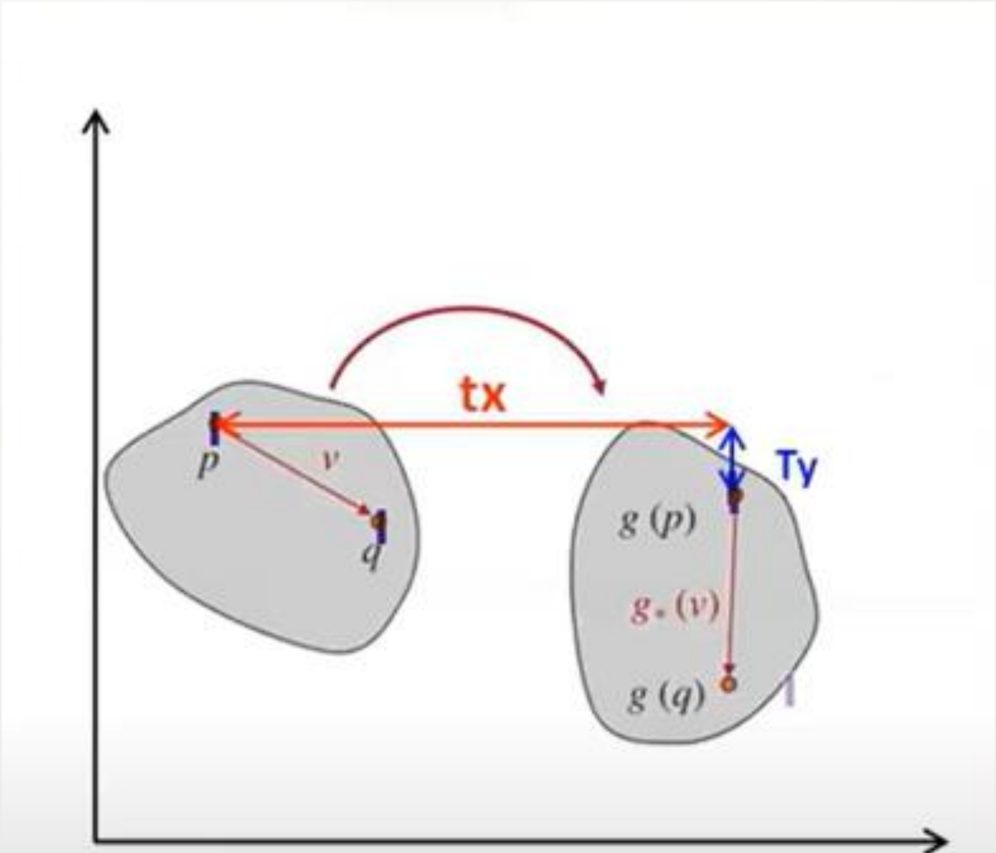
Types of Geometric Transformation
4. Shearing (Cont.)



Types of Geometric Transformation
5. Rigid Transformation

- Rigid = Translations + Rotations

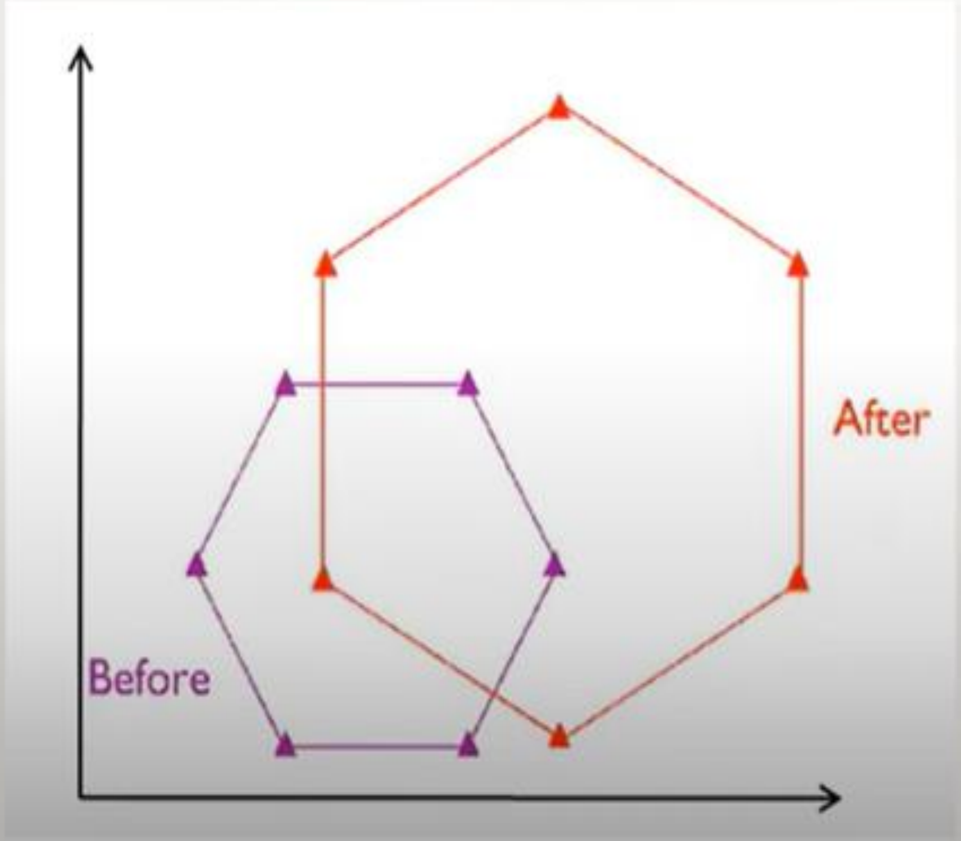


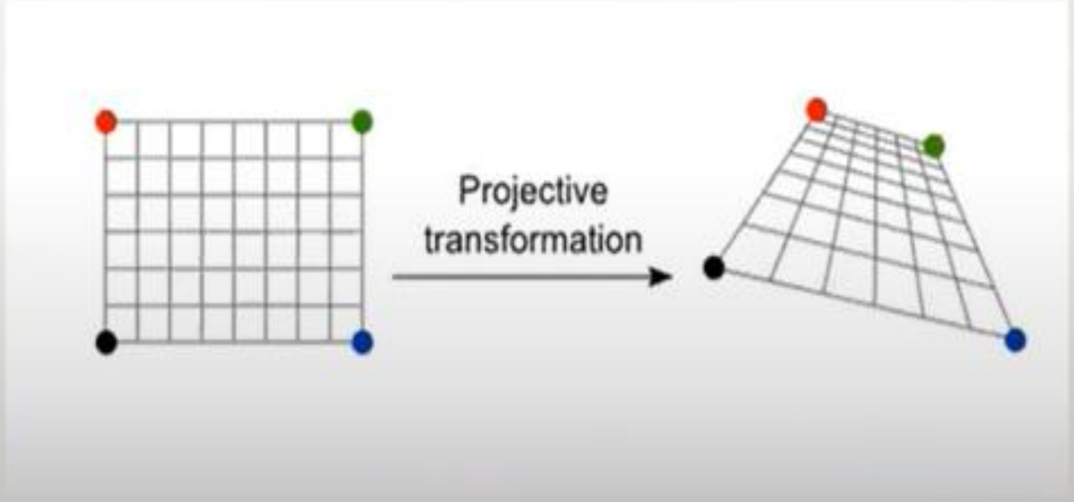


Types of Geometric Transformation
6. Similarity Transformation

- Similarity = Translations + Rotations + Scale

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \text{where } s = \text{scaling}$$



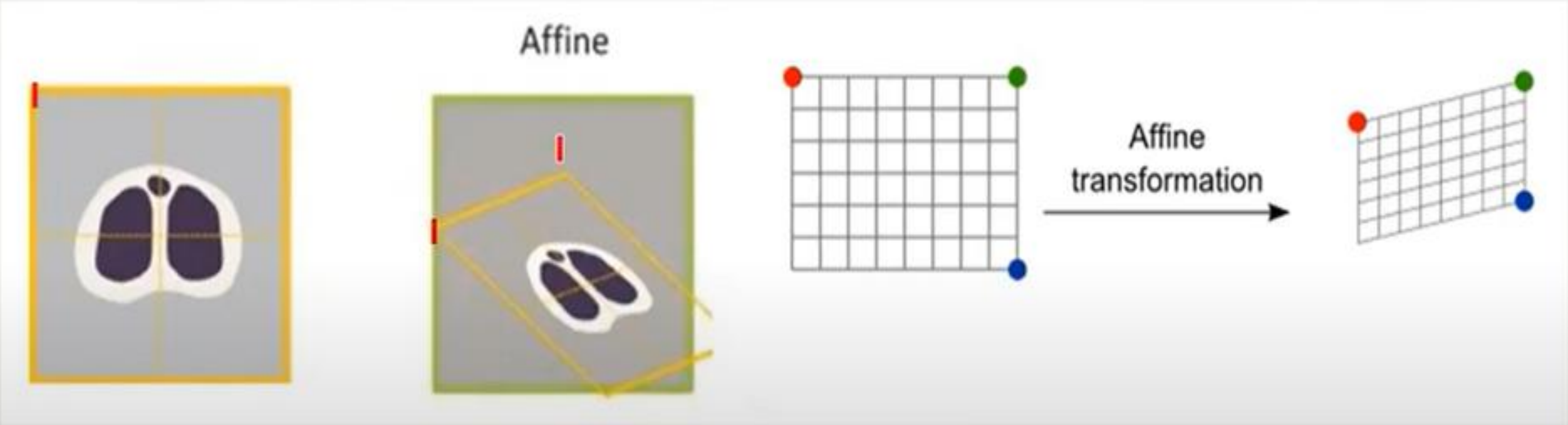


Types of Geometric Transformation
7. **Affine Transformation (IMP)**

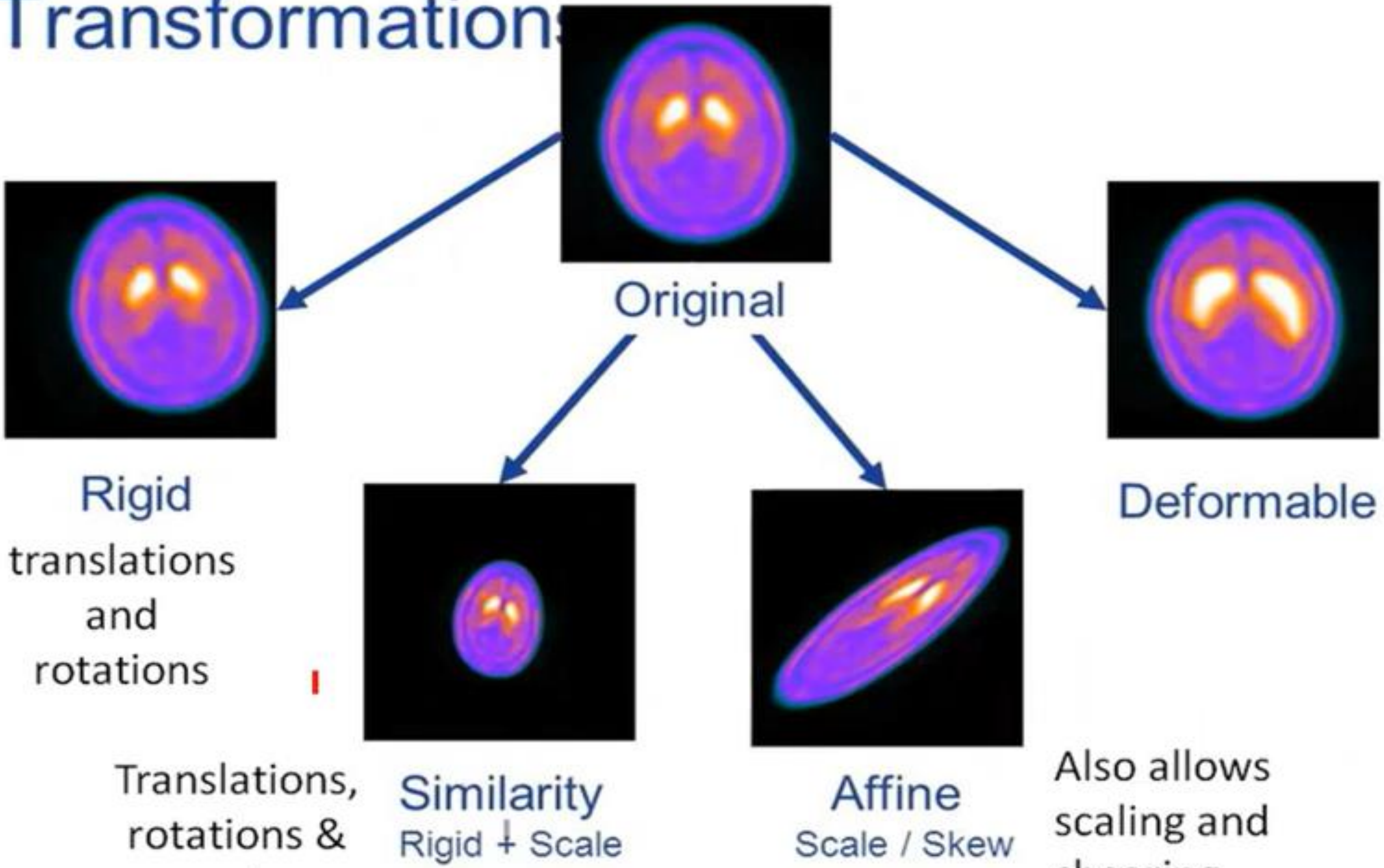
- Affine = Translations + Rotations + Scale + shear
- An affine transformation is a transformation that preserves co-linearity and the ratio of distances.
- The parallel lines in an original image will be parallel in the output image.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Types of Geometric Transformation
7. Affine Transformation (Cont.)



Example Registration Transformations

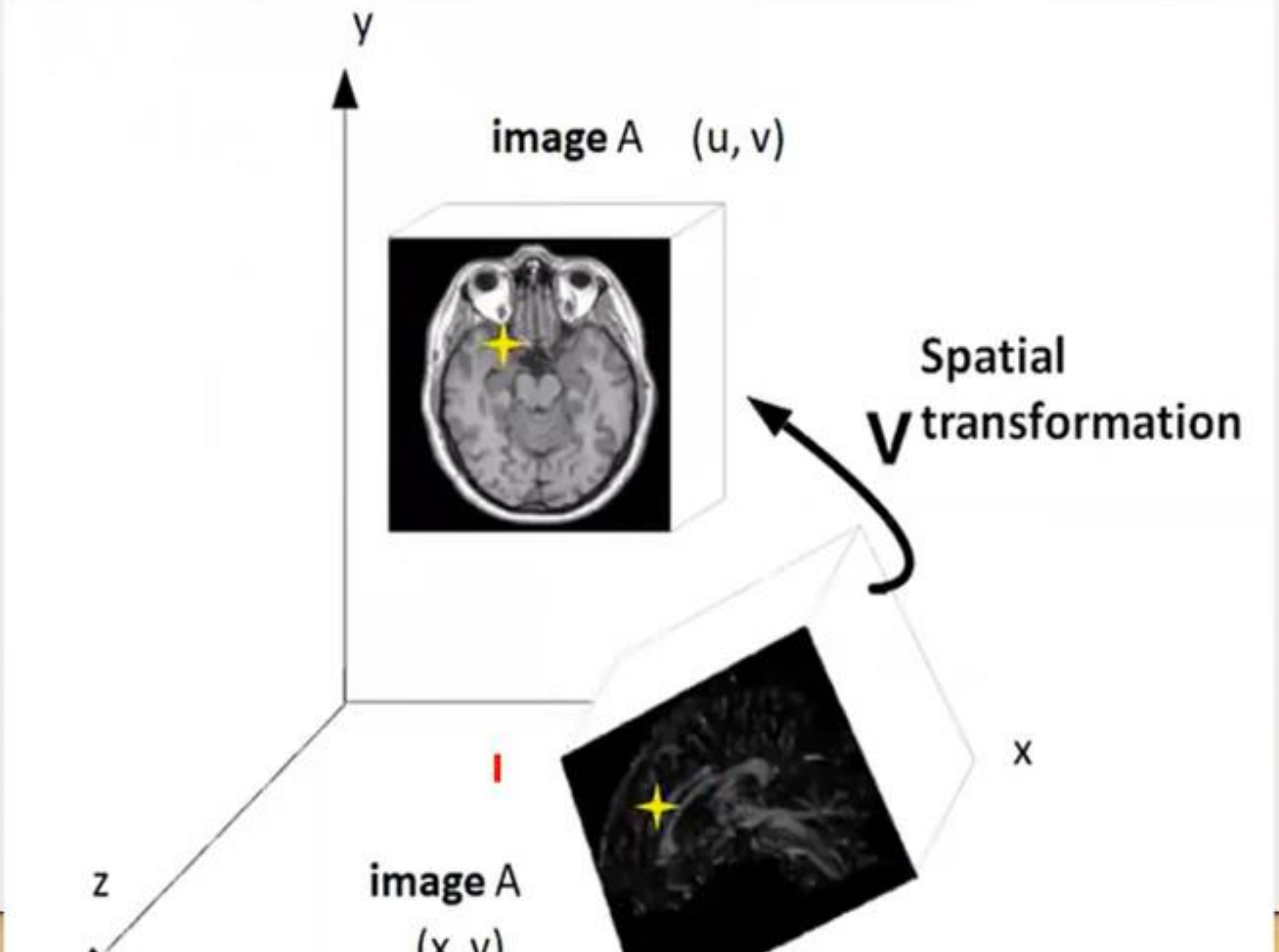


translations and rotations

Translations, rotations & scale

Also allows scaling and shearing

3D Spatial Transformation

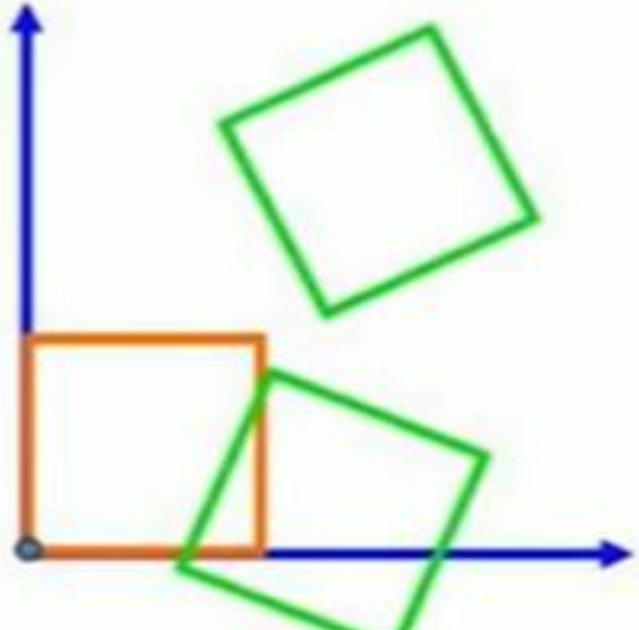


What is Euclidean or Isometric Transform ?

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$$

Properties

- $\mathbf{R}$ is an orthogonal matrix
- Euclidean distance is preserved
- Has three degrees of freedom; two for translation, and one for rotation



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}$$

Theta - Rotation we want to give

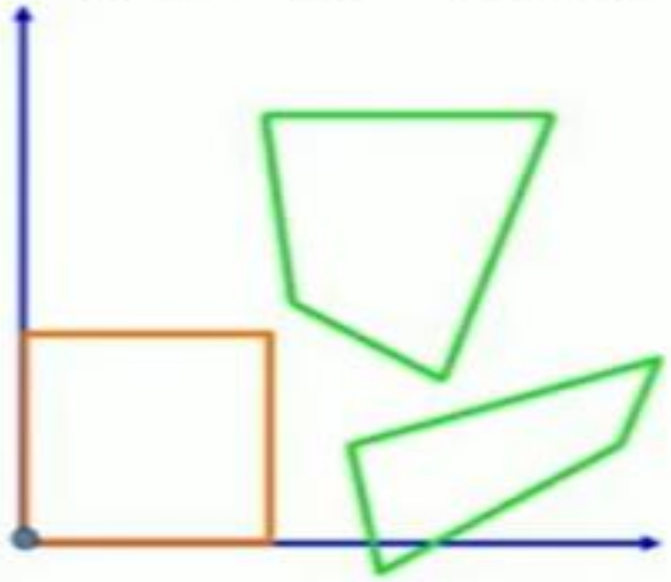
tx - Translation in X-axis

ty - Translation in Y-axis

Projective Transform

- Class IV: Projective transformation

$$c\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



Affine vs Projective Transform